International Journal of Internet, Broadcasting and Communication Vol.6 No.1 16-22 (2014)
http://dx.doi.org/10.7236/IJIBC.2014.6.1.16

IJIBC 14-1-5

Pattern Analysis and Performance Comparison of Lottery Winning Numbers

Yong Gyu Jung[1*], Soo Ji Han[2], Jae Hee kim[3]

[1*,2]*Dept. of Medical IT Marketing, Eulji University, Korea*
ygjung@eulji.ac.kr

[3]*Chief executive officer, T2L Corp, Korea*
toj@t2l.co.kr

Abstract

Clustering methods such as k-means and EM are the group of classification and pattern recognition, which are used in management science and literature search widely. In this paper, k-means and EM algorithm are compared the performance using by Weka. The winning Lottery numbers of 567 cases are experimented for our study and presentation. Processing speed of the k-means algorithm is superior to the EM algorithm, which is about 0.08 seconds faster than the other. As the result it is summerized that EM algorithm is better than K-means algorithm with comparison of accuracy, precision and recall. While K-means is known to be sensitive to the distribution of data, EM algorithm is probability sensitive for clustering .

Key words: *Lottery, gambling , k-means, EM Algoritm, Weka*

1. Introduction

In basic form a lottery is a popular form of gambling in which players draw lots from a pool to win a prize, in most cases, the prize being money. Lotteries, in one form or another, have been around as long as people have lived in society. The word lottery probably derives from the Italian word 'lottery' which means fate or destiny. However, in more recent history, lottery games have grown more complex and involved. It is said in the Bible that Moses used a lottery to distribute lands around the river Jordan. The popular game of Keno (Chinese Lottery) originated in ancient China around 3,000 years ago, originally set up by the government to raise funds for the construction of the Great Wall. Julius Caesar often used lotteries to raise funds for the building of his vast empire. One of the oldest lotteries still in operation today was begun in Portugal in 1783. In 1567 Queen Elizabeth I instituted the first English state Lottery. All of these early lotteries had similar purposes of raising funds for the government. Often governments would institute them in times of need or war. Much of the money used to build the universities Harvard, Yale, Princeton and Columbia was raised by lotteries. Today official lotteries are set up to give money back to the public and must somehow benefit the people.

Unfortunately, like many forms of gambling, lotteries are very prone to corruption. It is because that Paradigm that many people prefer to increase their property from investing money than saving money. Among the economic difficulties that changes frequently have dominated the economic life of the people . Under such circumstances, people who take expectations on lottery dreaming reversal of life. In fact, lottery sales ever since

Manuscript Received : Nov. 5, 2013 / Revised: Jan. 10, 2014 / Accepted: Feb.14, 2014
Corresponding Author: ygjung@eulji.ac.kr
Tel: +82-31-740-7178, Fax: +82-31-740-7178
Dept. of Medical IT Marketing, Eulji University, Korea

it reached a peak in 2003 have been reduced 10% per year. But, After the end of 2010 , the lottery sales was significantly increased. Also, interest about lottery has increased and research about lottery winning number regulrity has been progressing. In this paper, the performance of the cluster is compared using by the pattern analysis of the winning numbers from 567 cases. Weka is a popular suite of machine learning software written in Java, developed at the University of Waikato, New Zealand. Weka is free software available under the GNU General Public License. The data used is the fixed broadband Internet Users for 214 Countries from year 1998-2011. Fixed broadband Internet subscribers are the number of broadband subscribers with a digital subscriber line, cable modem, or other high-speed technology.

2. Related research

2.1 K-Means

The term *k*-means was first used by James MacQueen in 1967, though the idea goes back to Hugo Steinhaus in 1957. The standard algorithm was first proposed by Stuart Lloyd in 1957 as a technique for pulse-code modulation, though it wasn't published outside Bell labs until 1982. In 1965, E.W.Forgy published essentially the same method, which is why it is sometimes referred to as Lloyd-Forgy, too. A more efficient version was proposed and published in Fortran by Hartigan and Wong in 1975/1979.

Input *: The number of k and a database containing n objects*
Output *: A set of k clusters that minimize the squared-error criterion.*
Method *:*
(1) arbitarily choose k objects as the initial cluster centers
(2) repeat
(3) (re)assign each object to the cluster to which the object is the most similar
based on the mean value of the objects in the cluster.
(4) update the cluster mean, ie, calculate the mean
value of the object for each cluster
(5) until no change;

Fig. 1. k-means Algorithm

k-Means clustering is a method of vector quantization originally from signal processing, that is popular for cluster analysis in data mining. k-means clustering aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster. This results in a partitioning of the data space into Voronoi cells. The problem is computationally difficult (NP-hard); however, there are efficient heuristic algorithms that are commonly employed and converge quickly to a local optimum. These are usually similar to the expectation-maximization algorithm for mixtures of Gaussian distributions via an iterative refinement approach employed by both algorithms. Additionally, they both use cluster centers to model the data; however, k-means clustering tends to find clusters of comparable spatial extent, while the expectation-maximization mechanism allows clusters to have different shapes. The most common algorithm uses an iterative refinement technique. Due to its ubiquity it is often called the *k*-means algorithm; it is also referred to as Lloyd's algorithm, particularly in the computer science community.

Given an initial set of *k* means $m_1^{(1)},\ldots,m_k^{(1)}$ (see below), the algorithm proceeds by alternating between two steps:

Assignment step: Assign each observation to the cluster whose mean yields the least within-cluster sum of squares (WCSS). Since the sum of squares is the squared Euclidean distance, this is intuitively the "nearest" mean. (Mathematically, this means partitioning the observations according to the Voronoi diagram generated by the means).

$$S_i^{(t)} = \{x_p : \|x_p - m_i^{(t)}\|^2 \leq \|x_p - m_j^{(t)}\|^2 \ \forall\ 1 \leq j \leq k\},$$

where each x_p is assigned to exactly one $S^{(t)}$, even if it could be is assigned to two or more of them.

Update step: Calculate the new means to be the centroids of the observations in the new clusters.

$$m_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{x_j \in S_i^{(t)}} x_j$$

Since the arithmetic mean is a least-squares estimator, this also minimizes the within-cluster sum of squares (WCSS) objective.

The algorithm has converged when the assignments no longer change. Since both steps optimize the WCSS objective, and there only exists a finite number of such partitionings, the algorithm must converge to a (local) optimum. There is no guarantee that the global optimum is found using this algorithm.

The algorithm is often presented as assigning objects to the nearest cluster by distance. This is slightly inaccurate: the algorithm aims at minimizing the WCSS objective, and thus assigns by "least sum of squares". Using a different distance function other than (squared) Euclidean distance may stop the algorithm from converging. It is correct that the smallest Euclidean distance yields the smallest squared Euclidean distance and thus also yields the smallest sum of squares. Various modifications of k-means such as spherical k-means and k-medoids have been proposed to allow using other distance measures.

2.2 EM

The EM algorithm was explained and given its name in a classic 1977 paper by Arthur Dempster, Nan Laird, and Donald Rubin. They pointed out that the method had been "proposed many times in special circumstances" by earlier authors. In particular, a very detailed treatment of the EM method for exponential families was published by Rolf Sundberg in his thesis and several papers following his collaboration with Per Martin-Löf and Anders Martin-Löf. The Dempster-Laird-Rubin paper in 1977 generalized the method and sketched a convergence analysis for a wider class of problems. Regardless of earlier inventions, the innovative Dempster-Laird-Rubin paper in the *Journal of the Royal Statistical Society* received an enthusiastic discussion at the Royal Statistical Society meeting with Sundberg calling the paper "brilliant". The Dempster-Laird-Rubin paper established the EM method as an important tool of statistical analysis. The convergence analysis of the Dempster-Laird-Rubin paper was flawed and a correct convergence analysis was published by C. F. Jeff Wu in 1983. Wu's proof established the EM method's convergence outside of the exponential family, as claimed by Dempster-Laird-Rubin.

Input *: Cluster number k, a database, Stopping tolerance* $\varepsilon (> 0)$

Output *: A set of k clusters with weight that maximize Log-likelihood function.*

(1) Expectation Step

For each database record x,

Compute the membership probability of x in each cluster h = 1,...,k.

(2) Maximization Step

Update mixture model parameter (probability weight)

(3) Stopping criteria

If stop criteria is satisfied stop

Else set j = j+1 and goto (1)

Fig. 2. EM Algorithm

In statistics, an expectation–maximization (EM) algorithm is an iterative method for finding maximum

likelihood or maximum a posteriori (MAP) estimates of parameters in statistical models, where the model depends on unobserved latent variables. The EM iteration alternates between performing an expectation (E) step, which creates a function for the expectation of the log-likelihood evaluated using the current estimate for the parameters, and a maximization (M) step, which computes parameters maximizing the expected log-likelihood found on the *E* step. These parameter-estimates are then used to determine the distribution of the latent variables in the next E step.

EM is frequently used for data clustering in machine learning and computer vision. In natural language processing, two prominent instances of the algorithm are the Baum-Welch algorithm (also known as *forward-backward*) and the inside-outside algorithm for unsupervised induction of probabilistic context-free grammars. In psychometrics, EM is almost indispensable for estimating item parameters and latent abilities of item response theory models. With the ability to deal with missing data and observe unidentified variables, EM is becoming a useful tool to price and manage risk of a portfolio. The EM algorithm (and its faster variant Ordered subset expectation maximization) is also widely used in medical image reconstruction, especially in positron emission tomography and single photon emission computed tomography. See below for other faster variants of EM.

A Kalman filter is typically used for on-line state estimation and a minimum-variance smoother may be employed for off-line or batch state estimation. However, these minimum-variance solutions require estimates of the state-space model parameters. EM algorithms can be used for solving joint state and parameter estimation problems.

Filtering and smoothing EM algorithms arise by repeating the following two-step procedure.

E-Step

Operate a Kalman filter or a minimum-variance smoother designed with current parameter estimates to obtain updated state estimates.

M-Step

Use the filtered or smoothed state estimates within maximum-likelihood calculations to obtain updated parameter estimates.

Suppose that a Kalman filter or minimum-variance smoother operates on noisy measurements of a single-input-single-output system. An updated measurement noise variance estimate can be obtained from the maximum likelihood calculation

$$\hat{\sigma}_v^2 = \frac{1}{N}\sum_{k=1}^{N}(z_k - \hat{x}_k)^2$$

where $\hat{x}_k$are scalar output estimates calculated by a filter or a smoother from N scalar measurements z_k. Similarly, for a first-order auto-regressive process, an updated process noise variance estimate can be calculated by

$$\hat{\sigma}_w^2 = \frac{1}{N}\sum_{k=1}^{N}(\hat{x}_{k+1} - \hat{F}\hat{x}_k)^2$$

where $\hat{x}_k$and $\hat{x}_{k+1}$are scalar state estimates calculated by a filter or a smoother. The updated model coefficient estimate is obtained via

$$\hat{F} = \frac{\sum_{k=1}^{N}(\hat{x}_{k+1} - \hat{F}\hat{x}_k)}{\sum_{k=1}^{N}\hat{x}_k^2}.$$

The convergence of parameter estimates such as those above are studied in

3. Experiment

Weka for our experiment, which is a software developed in the java language in waikato university for data mining and machine learning. The Weka contains a collection of visualization tools and algorithms for data analysis and predictive modeling, together with graphical user interfaces for easy access to this functionality. Weka supports several standard data mining tasks, more specifically, data preprocessing, clustering, classification, regression, visualization, and feature selection. All of Weka's techniques are predicated on the assumption that the data is available as a single flat file or relation, where each data point is described by a fixed number of attributes. Weka's main user interface is the Explorer, but essentially the same functionality can be accessed through the component-based Knowledge Flow interface and from the command line. There is also the Experimenter, which allows the systematic comparison of the predictive performance of Weka's machine learning algorithms on a collection of datasets. In this study, we will use the version WEKA3.6.10 and we want to compare the performance of the EM algorithm with K-means algorithm by using statistical analysis file of Lottery 567 cases which is provided by the Lotterystyle. Winning numbers from 567 cases are prepared for our study. And they can be compared the clustering divided into four parts.

```
=== Run information ===

Scheme:weka.clusterers.EM -I 100 -N 4 -M 1.0E-6 -S 100
Relation:     통합 문서4
Instances:    6
Attributes:   567
[list of attributes omitted]
Test mode:evaluate on training data

=== Model and evaluation on training set ===


EM
==

Number of clusters: 4


            Cluster
Attribute         0       1       2       3
              (0.33)  (0.17)  (0.33)  (0.17)
============================================
1
  mean            31      40    16.5      37
  std. dev.        2 10.9301     6.5 10.9301

2
  mean            23      42      11      32
  std. dev.        2 12.1929       2 12.1929
```

Figure 1. EM training set

```
XMeans
======
Requested iterations            : 1
Iterations performed            : 1
Splits prepared                 : 1
Splits performed                : 0
Cutoff factor                   : 0.5
Percentage of splits accepted
by cutoff factor                : 0 %
------
Cutoff factor                   : 0.5
------

Cluster centers                 : 2 centers

Cluster 0
            34.75 30.0 24.5 35.75 38.0 33.75 26.75 33.75 27.0 37.0 3
Cluster 1
            16.5 11.0 13.5 20.5 20.0 14.5 5.5 13.5 3.0 17.0 4.0 6.5 2

Distortion: 27.228406
BIC-Value : -4256.991073
```

Figure 2. K-Means training set

4. Experimental Result

As the result of EM experiment, spending time to construct is 0.11 second and clustered instances are included two (33%) in cluster0, one (17%) in cluster1 and one (17%) cluster 3 shown in Figure 3.

```
Time taken to build model (full training data) : 0.11 seconds

=== Model and evaluation on training set ===

Clustered Instances

0       2 ( 33%)
1       1 ( 17%)
2       2 ( 33%)
3       1 ( 17%)


Log likelihood: -1489.88318
```

Figure 3. execution time of EM, and a cluster instance

While the result of K-Means experiment, spending time to construct is 0.03 second and clustered instance are included four (67%) in cluster0 and two (33%) in cluster1 shown in Figure 4.

```
Time taken to build model (full training data) : 0.03 seconds

=== Model and evaluation on training set ===

Clustered Instances

0       4 ( 67%)
1       2 ( 33%)
```

Figure 4. execution time of K-Means , and a cluster instance

5. Conclusion

Clustering is the fundamental task in data mining. It is a technique by which we can categorize between the similar and the dissimilar objects and group the ones together that are more likely to each other. Clustering is the process applied on datasets to partition the data into various meaningful subsets, called as the Clusters. The objects within each clusters share a common trait. The goal of the Cluster Analysis is descriptive and aims to discover a new set of categories. Clustering is about finding the similarity and to find how similar the two objects, the distance measure is used. The objects that are within the same cluster need to be close to each other. Hence, in case of the similar objects the distance measure will be a short distance.

In Figure 3 and 4, the features of experimental data is easy compared by showing a monotonic results. In other words, K-Means algorithm in Figure 3 were classified in the value of the property of the winning numbers of 567 cases on the basis of cluster 1 and cluster 0. But EM algorithm in Figure 4 was analyzed from cluster 0. We found that K-Means algorithm is 0.08 seconds faster than the EM algorithm. It means that K-means algorithm is useful in visible simple values. But k-means algorithm have a weakness for adaptability of outler, because it is sensitive about distribution of data.

References

[1] Olson, Carl M. "Lottery gaming apparatus and method." U.S. Patent No. 6,080,062. 27 Jun. 2000.
[2] Joung, Je-Gun, et al. "Identification of regulatory modules by co-clustering latent variable models: stem cell differentiation." Bioinformatics 22.16 (2006): 2005-2011
[3] Luk, Andrew, and S. Lien. "Learning with lottery-type competition." Neural Networks Proceedings, 1998. IEEE World Congress on Computational Intelligence. The 1998 IEEE International Joint Conference on. Vol. 2. IEEE, 1998.
[4] Eibe Frank, Ian H. Witten , Data Mining : Practical Machine Learning Tools and Techniques, Morgan Kaufmann Publishers , 2011
[5] Jongwan Kim , Improving Artificial Intelligence Lecture using WEKA Tool , Proceedings of KIIS Fall Conference 2012 Vol. 22, No. 2.
[6] Yong-Gyu Jung , Jun Heo, Kyu Ho kim, Using Discretization of Numeric Attributes to Compare the Changes in Performance of C4.5 and CART algorithms, International Conference of the Korea Distribution Science Association, ISSN2287-478X Vol.4 pp353-358 2013.7.11,